IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application No.: 10/729,331	§	Examiner: Fennema, Robert E.
Filed: December 05, 2003	§	Group/Art Unit: 2183
Inventor:	§	Atty. Dkt. No: 5500-91700
Teik-Chung Tan	§	
Gregory William Smaus	§	
	§	
	§	
Title: Multiple Control Sequences per Row of Microcode ROM	§	
	§	
	§	
	§	
	§	
	§	

REPLY BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This brief is in reply to the Examiner's Answer mailed August 22, 2007. Appellants respectfully request that this Reply Brief be entered pursuant to 37 C.F.R. § 41.41 and considered by the Board of Patent Appeals and Interferences.

REPLY

First ground of rejection:

Claims 1-3, 5-10, 12-14, 16-21, 23, 24 and 27 stand finally rejected under 35 U.S.C. § 102(b) as being anticipated by Tredennick et al. (U.S. Patent 4,338,661) (hereinafter "Tredennick"). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 1, 2, 12, 13, and 23:

Regarding independent claim 1, contrary to the Examiner's assertion, Tredennick clearly fails to disclose *a microcode ROM, wherein a row in the microcode ROM stores a plurality of groups of microcode operations, wherein a group of the plurality of groups of microcode operations is comprised in a microcode routine, and wherein the row stores an associated control sequence for each of the plurality of groups of microcode operations*.

Appellants' arguments from the Appeal Brief filed March 5, 2007 regarding the rejection of this claim are herein incorporated by reference. As explained in Appellants' Appeal Brief, the Examiner submitted that Tredennick teaches *wherein a row in the microcode ROM stores a plurality of groups of microcode operations*, using an example in which one row in the nano control of Tredennick includes four nanowords: swap1, swap2, tasm1, and tasm2, and asserted, "the groups could contain a single instruction, with some rows containing up to 4 groups of operation, based on the particular row and exact length of ROM chosen to implement the design." Appellants argued that a single instruction is not "a group of microcode operations (plural) ... comprised in a microcode routine" as recited in claim 1, and having the additional limitations of claim 1 discussed below. Appellants pointed out that claim 1 recites not just "wherein in response to accessing the group of microcode operations comprised in the microcode routine..." but

also, "the control sequence associated with the group of microcode operations..." clearly reciting that the group includes multiple microcode operations.

In his Answer, the Examiner submits that Appellants disregarded his actual rejection in their arguments above by focusing on his example of a "single instruction." However, Appellants addressed the remaining limitations referenced above (*wherein the row stores an associated control sequence for each of the plurality of groups of microcode operations*) in subsequent remarks, summarized below.

As discussed in Appellants' Appeal Brief, Appellants assert that Tredennick clearly fails to teach or suggest *wherein the row stores an associated control sequence for each of the plurality of groups of microcode operations*. The Examiner cited column 15, lines 52-55 and 59-66 as teaching this limitation. However, these passages clearly do not describe a control sequence being **associated with a group of *multiple* instructions** comprised in a row in the microcode, as in Appellants' claimed invention. Instead, each of the individual microwords corresponding to these nanowords includes a micro ROM base address for the next individual microword and nanoword to be accessed.

In his Answer, the Examiner submits that Appellants have validated Examiner's rejection of the claim in the remarks above. The Examiner notes that the claim is a *comprising* claim, such that the microprocessor must contain the elements, but does not preclude other elements from being present. The Examiner submits, "Tredennick simply has multiple control sequences associated with each group, which clearly fits the claim language of the group having an associated control sequence, in the comprising language." Appellants respectfully disagree and note that claim 1 includes the additional limitation *wherein in response to accessing the group of microcode operations comprised in the microcode routine, the control sequence logic unit is configured to use the control sequence associated with the group of microcode operations to identify another row storing one or more next groups of microcode operations comprised in the microcode routine.* The plain wording of this limitation of claim 1 requires that a single control

sequence (the control sequence) associated with the group of microcode operations referred to in the previous limitation is used to identify another row in response to accessing the group of microcode operations. Appellants again assert that Tredennick does not teach a control sequence associated with a group of operations, nor one used to identify another row in response to accessing the group of operations but instead teaches a micro ROM base address associated with an individual microword, used to address the next microword and nanoword in response to access of that individual microword, regardless of its inclusion in a group of such microwords.

The Examiner also cited column 15, lines 37-40 as teaching the limitation above ("it selects the next line of the ROM, which is output from the microcode ROMs as shown in column 15, lines 52-55 and 59-66.") However, as discussed above and in Appellants' Appeal Brief, Tredennick does not disclose a single control sequence (the control sequence) **associated with a group of microcode operations**. Instead, for each individual operation, a next address is provided for a next individual operation, if any, in the microcode routine.

In his Answer, the Examiner notes that, as shown in Tredennick's FIG. 11a, every address specifies a row, using the high 8 bits, "Therefore, every control sequence in Tredennick teaches a row where the next instruction (which is in a group, thus specifies a group) resides." First, Appellants note that not every "next instruction" in a row is included in a group, as the Examiner suggests. In addition, the Examiner is ignoring the rest of this limitation, which does not recite identifying a row of a "next instruction," but identifying a *row storing one or more next groups of microcode operations comprised in the microcode routine* (i.e., the same microcode routine.)

In his Answer, the Examiner also states, "Appellant seems to be arguing that when executing a program (microcode routine), that Tredennick somehow can only use the instructions on a single row, and not include other instructions in the microcode routine, which is clearly not the case, as seen above." **The Examiner has misinterpreted Appellants' argument.** Appellants' argument is that (as required by the

claims) one row in the microcode ROM includes multiple groups of operations, that one of these groups of operations is part of a microcode routine, that the row includes a control sequence associated with each group of operations, and that the control sequence associated with the group of operations in the microcode routine identifies another group of operations on another row that are also part of the same microcode routine. This is clearly not taught by Tredennick.

Appellants also note that the individual microwords in Tredennick include a micro ROM base address for the next individual microword and nanoword to be accessed, only if there are other microwords and nanowords to be accessed within the currently executing microcode routine. The last microword in each microcode routine does not include a next micro ROM address field. For example, swap2 (the last microword in a microcode routine) does not include a value in the NMA field in Appendix A. Therefore, contrary to the Examiner's suggestion, neither of the operations making up the microcode routine in his example using swap1 and swap2 includes an associated control sequence to identify additional operations for that microcode routine on another row. Instead, the next microword accessed will depend on the decoding of the next macroinstruction (in the computer program). This will initiate the execution of a next microcode routine, which is not associated in the control store with the microwords and nanowords making up the microcode routine that implemented the previous macroinstruction.

In his Answer, the Examiner goes on to say, "When the swap or trap instructions are complete, some other instruction will be executed which is on a different row. It appears Appellant is trying to read detail from his specification into the claims as to what a microcode routine is (and Appellant seems to be implying that a microcode routine is a single instruction, which is not the common usage of the term), when it has not been defined in any way in the claims, and has been interpreted by the Examiner in the broadest reasonable way, which is a computer program" (emphasis added). **Appellants assert that the term "microcode routine", as well as alternate forms "microinstruction routine" and "microprogram", are terms of art well known by**

those skilled in the art, especially in the context of microcoded instructions within an integrated circuit (as in the present invention), and could not be interpreted as merely "a computer program" by anyone having benefit of Appellants' disclosure, or the disclosure of Tredennick, for that matter.

For example, Tredennick states, in column 1, "This invention relates generally to data processors and more particularly to a data processor having a control store containing a plurality of microinstruction routines for implementing instructions received by the data processor" (emphasis added). Similarly, column 2 of Tredennick includes the following, "Generally, a data processor performs a series of operations upon digital information in an execution unit according to a stored program of instructions. These instructions are often termed "macroinstructions" in order to avoid confusion with the microinstructions contained in the control store of the data processor. The microinstructions are often grouped into microinstruction routines in order to perform various macroinstructions" (emphasis added). The descriptions and figures cited by the Examiner in Tredennick are directed to the implementation and contents of this control store (i.e., the combination of the micro ROM and nano ROM) and the use of these microinstruction routines to implement program instructions of a computer program (i.e., macroinstructions, using Tredennicks' terminology).

Similar descriptions of microcode routines stored in a microcode ROM, each of which corresponds to a microcoded instruction, are included in Appellants' specification, and no other use of the term "microcode routine" is described therein. Appellants respectfully remind the Examiner that, according to 2111 of the M.P.E.P., the Patent and Trademark Office ("PTO") determines the scope of claims in patent applications not solely on the basis of the claim language, but upon giving claims their broadest reasonable construction "in light of the specification as it would be interpreted by one of ordinary skill in the art." *In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364[, 70 USPQ2d 1827] (Fed. Cir. 2004). The broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach. *In re Cortright*, 165 F.3d 1353, 1359, 49 USPQ2d 1464, 1468 (Fed. Cir. 1999).

Appellants assert that no one skilled in the art having benefit of Appellants' disclosure and the cited art would reach the interpretation that the "microcode routine" of Appellants' claims is merely a "computer program." **Therefore, the Examiner's reference to "some other instruction" that would be executed following a trap or swap instruction have nothing to do with the claimed limitations involving groups of operations on different rows that are part of a (single) microcode routine.** In addition, as discussed above, there is no Next Micro Address associated with the last operation of a microcode routine identifying "the next instruction" as the Examiner suggests. Instead, the microcode routine implementing the next instruction is determined by decoding the next macroinstruction in the computer program.

In his Answer, the Examiner also refers to Appendix A and column 17. The Examiner submits, "it can be seen that there is a listing of destinations of each of these instructions, which point to a specific row to go to, which are not in the same row as the current instruction. For example, in Column 17, lines 29-32, a conditional branch is given as an example, with 2 possible outcomes. Both outcomes must be in the same row, however, this is not the same row as the branch is in, as can be seen in Appendix A." Appellants note that Appendix A does not include destinations or rows for each of the microinstructions listed, as the Examiner implies. Instead, destinations are listed only for microinstructions of a conditional branch type and rows are listed only for microinstructions that may be destinations of conditional branch type microinstructions. In addition, Appellants assert that the information included in Appendix A does not teach the specific limitations of Appellants' claim 1, i.e., that one row in the microcode ROM includes multiple groups of operations, that one of these groups of operations is part of a microcode routine, that the row includes a control sequence associated with each group of operations, and that the control sequence associated with the group of operations in the microcode routine identifies another group of operations on another row that are also part of the same microcode routine.

For at least the reasons above, Appellants again assert that Tredennick clearly does not teach all of the limitations of Appellants' claim 1. Therefore, the rejection of claim 1 is unsupported by the cited art and removal thereof is respectfully requested.

Independent claims 12 and 23 include limitations similar to claim 1, and so the arguments presented above apply with equal force to these claims, as well.

Claim 27:

Regarding independent claim 27, contrary to the Examiner's assertion, Tredennick clearly fails to teach or suggest *a microcode ROM, wherein a row in the microcode ROM stores a plurality of groups of microcode operations and wherein the row stores an associated control sequence for each of the plurality of groups.* In the previous Office Action, the Examiner cited column 15, lines 33-34, 36-37, 52-55, and 59-66 as teaching these limitations. However, as discussed above regarding claim 1, Tredennick does not teach or suggest a row in the microcode storing groups of microcode operations and a control sequence associated with each of the groups.

Further regarding claim 27, Tredennick fails to teach or suggest *means for accessing a control sequence associated with one of the plurality of groups of microcode operations and responsively accessing a next group of microcode operations stored in the microcode ROM.* The Examiner again cited "Column 15, lines 37-40, it selects the next line of the ROM, which is output from the microcode ROMs as shown in column 15, lines 52-55 and 59-66." However, as discussed above regarding claim 1, Tredennick does not teach or suggest groups of microcode operations stored in a row that also stores a control sequence associated with one of the **groups** of microcode operations, and therefore, cannot teach or suggest accessing such a control sequence or responsively accessing a next **group** of microcode operations stored in the microcode ROM. Therefore, Tredennick cannot be said to anticipate claim 27.

In his Answer, the Examiner states, "However, as explained above, if each control sequence identifies an instruction, and each instruction is in a group, then the identification of the instruction identifies the group as well. Given that a group has not been given any kind of definition, Examiner feels that by accessing an instruction in a group, then the group was clearly accessed as well. Examiner also notes for Claim 27 that there is no explicit indication that a group contains multiple instruction." Appellants again assert that the plain language of the claim, interpreted in light of Appellants' specification, indicates that the plurality of groups of microcode operations recited in the claim refers to a plurality of groups, each including multiple microcode operations, and that a single control sequence is associated with each group of microcode operations. Therefore, as discussed above, Tredennick does not anticipate claim 27.

For at least the reasons above, the rejection of claim 27 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 3 and 14:

Regarding claim 3, contrary to the Examiner's assertion, Tredennick clearly fails to teach or suggest *wherein the control sequence logic unit is configured to identify which of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine based on information contained in the control sequence associated with the group of microcode operations stored in the row*. The Examiner cited column 15, lines 52-55 and 59-66, submitting, "which defines which row and position the next group is located," and "The cited portion discloses that the address directs the processor to the next group in the routine, which must specify a row, as discussed in Claim 1. It can be further seen in Figure 11a that the 10-bit address does select a position using bits A1 and A0." Appellants argued that this citation does not teach identifying a row and position for a next group of microcode operations. Instead, this passage describes how the next address is determined for an individual microword having a type I or type II format. The address selects a single microword or nanoword,

not a group of microcode operations stored in a row and comprised in the microcode routine.

In his Answer, the Examiner again submits, "that by identifying the address of the next instruction, which is in a group, that the group is identified as well, and thus fits the claim language." Appellants again assert that the plain language of the claims requires that a single control sequence is associated with a group of microcode operations (plural), which is not taught by Tredennick. Therefore, Appellants also assert that there is nothing in Tredennick that teaches that a control sequence logic unit identifies which of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the (same) microcode routine, as required by claim 3, only that it identifies a next (individual) operation.

For at least the reasons above, the rejection of claim 3 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 14 includes limitations similar to claim 3, and so the arguments presented above apply with equal force to this claim, as well.

Claims 5, 16, and 24:

Regarding claim 5, contrary to the Examiner's assertion, Tredennick clearly fails to teach or suggest *wherein if the group of microcode operations comprises at least one branch operation, the control sequence logic unit is configured to identify the next group of microcode operations in the microcode routine dependent on a branch prediction as well as the control sequence associated with the group of microcode operations.* The Examiner cited column 15, lines 49-55 ("teach a microcode instruction for branches") and column 17, lines 29-32 ("show that the outcome either way will be in the same row specified by the control sequence") as teaching these limitations. Appellants argued that this citation does not describe identifying the next group of microcode operations in a microcode routine, but instead describes two alternate (individual) addresses that may be

the destination of a branch operation, and that the Examiner's remarks appear to teach away from a need to identify the next group of operations dependent on a branch prediction. **Appellants also asserted that branch prediction is not disclosed in Tredennick.**

In his Answer, the Examiner refers to his arguments regarding claim 3 "by identifying the address of the next microword, Tredennick identifies a row and a group in the process." Appellants again assert that the plain language of the claim requires that a single control sequence is associated with a group of microcode operations, which Tredennick does not teach. Therefore, Appellants also assert that there is nothing in Tredennick that teaches that a control sequence logic unit identifies the next group of microcode operations (plural) in the same microcode routine based (in part) on such a control sequence, as required by claim 5.

In his Answer, the Examiner also submits, "Examiner is taking the stance that because branch prediction has been such an integral and important part of modern processors, going back decades, and providing such a large and important performance increase, that Tredennick does utilize branch prediction, and it is not mentioned because it is such a basic and well-known concept that there is no reason to disclose it in a patent, because everyone does it and everyone knows what it is. With branch prediction resolution being one of the biggest latency problems with computers (along with memory access times), it would be foolish for one to not use branch prediction in a processor, and given that Tredennick already has outcomes for both the taken and not taken branches, Examiner believes that Tredennick makes use of branch prediction, because as stated earlier, branch prediction is one of the most basic and fundamental concepts of modern computer architecture." **Appellants assert that the Examiner's remarks are entirely speculative, with absolutely no basis in the cited reference.** The Examiner seems to imply that branch prediction is inherent in all microprocessors, including that of Tredennick, which is clearly not the case. Early microcoded processors without extensive pipelining, such as that of Tredennick, **did not employ branch prediction**. Because they did not include extensive pipelining, these processors would not have

achieved much, if any, performance improvements for all the added complexity that such mechanisms would have added to the architecture. Similarly, most RISC processors, as well as the original Pentium™, do not include branch prediction logic. For example, RISC processors are designed with an explicit goal of simplicity incompatible with such complex circuitry. Many other examples of processors without branch prediction mechanisms may be found in the field. **Clearly, the Examiner's assertions that "everyone does it" and that "it would be foolish for one to not use branch prediction in a processor" are insupportable.**

Even if "everyone knows what it is," not everyone included it in their designs, nor would branch prediction be appropriate or even obvious for all designs. In addition, Branch prediction is not even mentioned in Tredennick. Tredennick is directed to a "Conditional branch unit for microprogrammed data processor" (title) and includes extensive descriptions and figures illustrating the branching mechanism claimed, including the instruction sequencing logic and conditional branch control logic (branch control unit 80) of the data processor. For example, FIGs. 2, 3, 4, 7A-7D, 8, 9, 13, 14A-B, and 15 all include details about branching and instruction sequencing in the system of Tredennick. In addition, the section entitled "CONDITIONAL BRANCH LOGIC" in columns 22-26 describes the implementation of a multiplexer used to select one of a plurality of branch targets and the decoding logic used to control the output of the selected microwords and nanowords in the control store, dependent on the branch control logic. **Nothing in any of the figures, their descriptions, or anything else in Tredennick suggests or implies that branch prediction is employed or that it is used in identifying a next group of operations in a microcode routine.** It is not clear that it would even be possible to include a branch prediction mechanism in the data processor of Tredennick, given the operation of the microprogrammed control store and instruction sequencing mechanisms described. **In fact, it is Appellants' understanding that Tredennick is a patent for the Motorola 68000 microprocessor, and it is a well-known fact that the Motorola 68000 microprocessor does not employ branch prediction.**

Appellants again assert that Tredennick does not teach branch prediction at all, and therefore cannot (and does not) teach *wherein if the group of microcode operations comprises at least one branch operation, the control sequence logic unit is configured to identify the next group of microcode operations in the microcode routine dependent on a branch prediction as well as the control sequence associated with the group of microcode operations*, as recited in claim 5. **Without teaching branch prediction, and the specific limitations related thereto, Tredennick cannot be said to anticipate claim 5.**

For at least the reasons above, the rejection of claim 5 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 16 and 24 include limitations similar to claim 5, and so the arguments presented above apply with equal force to these claims, as well.

Claims 6 and 17:

Regarding claim 6, contrary to the Examiner's assertion, Tredennick clearly fails to teach or suggest *wherein the microcode ROM is divided into a plurality of segments, wherein a same number of groups of microcode operations is stored in each row of a given one of the plurality of segments, and wherein each row in the given one of the plurality of segments stores a different number of groups of microcode operations than each row in each other one of the plurality of segments*. The Examiner cited column 19, lines 22-27 as teaching this limitation, submitting, "for each row, the address may represent one, two, four, or up to eight different groups. So there are segments in the sense that some lines can contain a different number of groups than the other lines" and "in Tredennick, the word line in one ROM is a row (the micro ROM), or a 'group' in the nano ROM. Thus, when reading Tredennick as it was intended, a 'row' or 'word' in the micro ROM corresponds to multiple 'words' in the nano ROM." Appellants argued that the Examiner misquoted and misinterpreted column 19, asserting that in Tredennick, "a word line" is not a row in the microcode ROM that stores a plurality of groups of microcode operations, as in Appellants' claimed invention. Instead, each input address

identifies a single "word line" in the micro ROM and a single "word line" in the nano ROM. Each "word line" selects a single microword or a single nanoword from the micro ROM and nano ROM, respectively; however, more than one input address may identify the same nano word. In other words, each input address selects a single nano word, but multiple input addresses may select the same single nano word.

In his Answer, the Examiner submits, "Appellant is ignoring a crucial part of this citation, and is in fact attempting to twist the interpretation of this citation by emphasizing incorrect parts of the sentences. Now, looking at the Appellants original argument, which states that a word line can only select a single nanoword, it is abundantly clear that Tredennick states the exact opposite. Looking again at the citation, which clearly states: 'The line which remains high will cause the appropriate output value to be generated as the micro ROM output word and the nano ROM output word according to the coding at the intersection of the selected word line and output columns' (emphasis by Examiner). It is extremely clear from this citation that the output word is generated by the intersection of the word line and column, where the word line is the row specifying multiple operations." **Appellants traverse the Examiner's argument. The citation quoted above describes the decode circuitry of an exemplary PLA structure illustrated in FIG. 9 and used for selecting a single microword and a single nanoword in a control store. As described in more detail in column 18, lines 23-68, the decoding of the address bits (A_0-A_2) selects one of the eight word lines in the micro ROM and one of the four word lines in the nano ROM. As illustrated in FIG. 9, the circuitry at the intersections of the selected word line and the output columns determines the value of each bit of the selected output word, not which word is selected, as the Examiner suggests.**

The Examiner goes on to say, "Appellant has completely ignored the actual teachings of the section he cited, and is attempting to focus on different words of the citation to prove a statement that runs contradictory to the teachings of Tredennick. In fact, looking at the citation, there is no room for question or error that Tredennick teaches what the Examiner claims, that a word line corresponds to a **row**, and the selected output

word is **the intersection of the row and column**. Therefore, it is very clear that Appellant is incorrect in his argument, and Examiner is correct, and there can be no other interpretation of the cited section that says otherwise, as Tredennick is very clear on what a word line is." **Appellants assert that it is the Examiner who has taken the citation out of the context of the description of FIG. 9 (beginning at column 18, line 23) to misinterpret the teachings of Tredennick**. Appellants again note column 19, line 22-23, "Each word line in the micro ROM is represented by only one input address." In the simplified example illustrated by FIG. 9, the addresses in the control store include only 3 bits, and each combination of these three bits selects a single word line in the micro ROM and the nano ROM (though more than one combination may select the same single word line). There is no additional decoding illustrated or described to select a particular word from among a plurality of words selected by a word line, as would be required in the Examiner's interpretation. **Instead, output words are selected using only the word line.** The columns represent bits of the output words, whose values are determined by the circuitry at the intersections between the selected word line and these output columns.

Appellants also assert that even if the Examiner's assertion that a word line selects a row specifying multiple operations, which is clearly not the case, this does not teach the specific limitations of claim 6, *wherein the microcode ROM is divided into a plurality of segments, wherein a same number of groups of microcode operations is stored in each row of a given one of the plurality of segments, and wherein each row in the given one of the plurality of segments stores a different number of groups of microcode operations than each row in each other one of the plurality of segments*. **Nothing in Tredennick teaches these limitations, regardless of how the decode logic of the control structure is described.**

For at least the reasons above, the rejection of claim 6 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 17 includes limitations similar to claim 6, and so the arguments presented above apply with equal force to this claim, as well.

Claims 7-9 and 18-20:

Appellants assert that for reasons similar to those discussed above regarding claim 6, Tredennick fails to teach or suggest the limitations of claims 7-9. The Examiner cited column 19, lines 22-27 as teaching these limitations by the same reasoning applied to his rejection of claim 6. Appellants argued that this citation has nothing to do with a plurality of segments in a microcode ROM, much less with such segments having the limitations recited in claims 7-9, and that Tredennick teaches nothing about the maximum width of any such segments or what would be stored in such a segment (e.g., *one group of microcode operations and one associated control sequence per row*, as claimed.)

In his Answer, the Examiner submits that Appellant has argued that in the claimed invention, a row contains one group of microcode operations and one associated control sequence per row, which has not been claimed. The Examiner has misquoted Appellants' argument. Appellants were describing the contents of a segment, as recited in claim 9, "wherein one of the plurality of segments stores one group of microcode operations and one associated control sequence per row." The Examiner again quotes Tredennick, "Each word line in the micro ROM is represented by only one input address. Each word line in the nano ROM however may represent one, two, or four possible different input addresses. In the preferred embodiment of the data processor, a word line in the nano ROM may represent as many as eight different input addresses." The Examiner interprets this section 'in that each word line may contain a different number of operations, and the lines that contain the same number of operations fit into what has been claimed as a "segment", as they all have the same "maximum width."' **However, as discussed in Appellants' Appeal Brief, this section actually describes that while each input address selects a single nano word, multiple input addresses may select the same single nano word. It has absolutely nothing to do with the segments of Appellants' claims.**

For at least the reasons above, the rejection of claims 7-9 is unsupported by the cited art and removal thereof is respectfully requested. Claims 18-20 include limitations similar to claims 7-9 and so the arguments presented above apply with equal force to these claims, as well.

Claims 10 and 21:

Regarding claim 10, contrary to the Examiner's assertion, Tredennick clearly fails to teach or suggest *wherein the control sequence logic unit is configured to identify a position of one or more groups of microcode operations and a position of one or more control sequences dependent on which of the plurality of segments of the microcode ROM stores the one or more groups of microcode operations*. The Examiner cited column 15, lines 52-55 and 59-66, "which defines which row and position the next group is located," as teaching this limitation. Appellants argued that this citation this citation does not teach identifying a row and position for a next group of microcode operations. Instead, this passage describes how the next address (which is clearly not the same as "a row and position") is determined for an individual microword having a type I or type II format. Tredennick does not teach or suggest a microcode ROM organized according to the rows, routines, groups, and segments of Appellants' claimed invention, and clearly does not teach identifying a position of one or more groups of microcode operations and control sequences dependent on which of a plurality of segments stores the one or more groups of microcode operations. Therefore, Tredennick clearly does not anticipate claim 10.

Appellants note that, in his Answer, the Examiner did not include any additional remarks directed specifically to claim 10.

For at least the reasons above, the rejection of claim 10 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 21 includes limitations similar to claim 10, and so the arguments presented above apply with equal force to this claim, as well.

Second ground of rejection:

Claims 4, 11, 15, 22, and 25-26 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Tredennick in view of Yoshida (U.S. Patent 5,761,470). Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Claims 4, 15, and 25:

Regarding claim 4, contrary to the Examiner's assertion, Tredennick in view of Yoshida clearly fails to teach or suggest *wherein if fewer than all of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine, the control sequence logic unit is configured to substitute NOPs for the microcode operations comprised in the groups not comprised in the microcode routine when outputting the row to the scheduler.* The Examiner submitted that Tredennick teaches the microprocessor of claim 3, which Appellants traverse above. The Examiner admitted that Tredennick fails to teach the above-referenced limitation of claim 4, and relies on Yoshida to teach it.

Appellants' arguments from the Appeal Brief filed March 5, 2007 regarding this rejection are herein incorporated by reference. As explained in Appellants' Appeal Brief, Yoshida teaches of a VLIW machine, which exploits parallelism by executing multiple instructions simultaneously. Yoshida teaches that if the conventional VLIW machine cannot execute an instruction from the word in parallel, it inserts a NOP in its place, as it has to execute some instruction (column 1, lines 51-56). The Examiner submitted that, "Given the advantage of higher speed though parallelism, one of ordinary skill in the art at the time the invention was made would have converted Tredennick's invention to operate in a parallel fashion such as a VLIW machine to increase the speed and performance." Appellants argued that converting Tredennick's invention to operate in a parallel fashion would clearly (and dramatically) change the principle of operation of his

invention, and would, therefore, not be obvious to one of ordinary skill in the art at the time the invention was made. Appellants also argued that converting Tredennick's invention "to operate in parallel fashion" would not necessarily result in Appellants' claimed invention, as Yoshida does not describe the organization of a microcode ROM at all, much less one in which groups of microcode operations are stored in particular rows. Appellants also argued that all parallel processors necessarily include this "feature" of Yoshida, nor does the reference (or the Examiner, in his remarks) explain how this feature would be implemented in Tredennick's processor if "converted to operate in parallel fashion."

The Examiner further argued that Yoshida teaches "the need to insert NOP's in place of instructions fed to the machine that could not be executed". Appellants argued that Yoshida teaches inserting NOP's where no operations are available to be fed to one of the machines (i.e., when there are no available operations that can be executed in parallel.) **Appellants argued that this feature does not teach the limitations of claim 4, which recites substituting NOPs for microcode operations of another microcode routine when a row is output to the scheduler.** Yoshida clearly does not teach substituting NOPs for microcode operations so that they will not be output to a scheduler, nor would including Yoshida's feature (feeding NOPs to one data processor with nothing to execute when another data processor executing in parallel does have an operation to execute) in a modified invention of Tredennick teach this limitation.

In his Answer, the Examiner submits that "Yoshida would not require as much effort to be combined with Tredennick as other references may, because they are analogous in several ways. Namely, a VLIW word is essentially a long instruction word with multiple instructions on it, which is very similar to the nano ROM word of Tredennick's invention. The entire line could be grabbed and used as a VLIW word. Yoshida's Figure 1 shows that there is a fair amount of similarity between the two inventions." **Appellants traverse the Examiner's argument. First, as discussed in detail above, the nano ROM word (or word line) of Tredennick does not include multiple instructions on it, as the Examiner suggests. Instead, it includes the output**

bits of a single nano word, which is used to execute one operation of a microcode routine. Furthermore, merely grabbing a VLIW word from the control store of Tredennick would not be sufficient to use the teachings of Yoshida in the system of Tredennick. There would need to be extensive changes to parse the different instructions of the VLIW, to distribute them to different execution units, etc. Appellants again assert that such a massive change renders the combination non-obvious.

In his Answer, the Examiner also disagrees with Appellants assertion that the combination of references would not teach the claimed invention. The Examiner submits that Yoshida's teaching that NOPs are inserted where no operations are available to be executed in parallel fits the claim language "*wherein if fewer than all of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine, the control sequence logic unit is configured to substitute NOPs for the microcode operations comprised in the groups not comprised in the microcode routine when outputting the row to the scheduler.*" **Appellants traverse.** First, the criteria for including NOPs are different between the two. Yoshida teaches inserting NOPs when operations within a VLIW word are not able to be executed in parallel. By contrast, claim 4 recites substituting NOPs *if fewer than all of a plurality of groups of microcode operations stored in the other row of the microcode ROM* (i.e., the other row indicated by the control sequence of a previous group of operations in a microcode routine) *are comprised in the* (same) *microcode routine*. In addition, Yoshida inserts NOPs in place of operations in the same VLIW word (which represents a single instruction in a computer program), whereas claim 4 recites substituting NOPs for microcode operations in groups of operations not in the same microcode routine (i.e., operations included in a microcode routine for a different instruction). **Appellants assert that these are clearly not the same at all.** Appellants again assert that that the Examiner has not provided a proper motivation to combine the references, and that even if combined, Tredennick in view of Yishida fails to teach or suggest all the limitations of claim 4.

For at least the reasons above, the rejection of claim 4 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 15 and 25 include limitations similar to claim 4, and so the arguments presented above apply with equal force to these claims, as well.

Claims 11, 22, and 26:

For reasons similar to those discussed above regarding claim 4, Tredennick in view of Yoshida clearly fails to teach or suggest *wherein a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine and are output during a single access*, as recited in claim 11. The Examiner admitted that Tredennick fails to teach this limitation, and again relies on Yoshida's VLIW machine to teach this limitation. Appellants again argued that the Examiner's proposed modification of the prior art would change the principle of operation of the prior art invention being modified, and thus the teachings of the references are not sufficient to render the claims *prima facie* obvious.

In addition, Appellants argued that the Examiner's citations in Yoshida do not teach *a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine and are output during a single access*. They teach that one VLIW word can specify a plurality of instructions, but they do not disclose that instructions are output during a single access, that they are stored in rows, or wherein a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in a microcode routine. Appellants also argued that Yoshida cannot suggest accessing multiple groups of the groups in the row of Tredennick because Yoshida does not teach accesses multiple groups of groups of operations itself or multiples ones of the VLIW instructions. Instead it teaches accessing a single instruction (which may be said to include a single group of operations) at a time. Appellants assert that even if Tredennick were combined with the referenced features of Yoshida, it would not teach the limitations of claim 11.

Appellants note that, in his Answer, the Examiner did not include any additional remarks directed specifically to claim 11.

For at least the reasons above, the rejection of claim 11 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 22 and 26 include limitations similar to claim 11, and so the arguments presented above apply with equal force to these claims, as well.

CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-27 is erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge any fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5500-91700/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: October 19, 2007